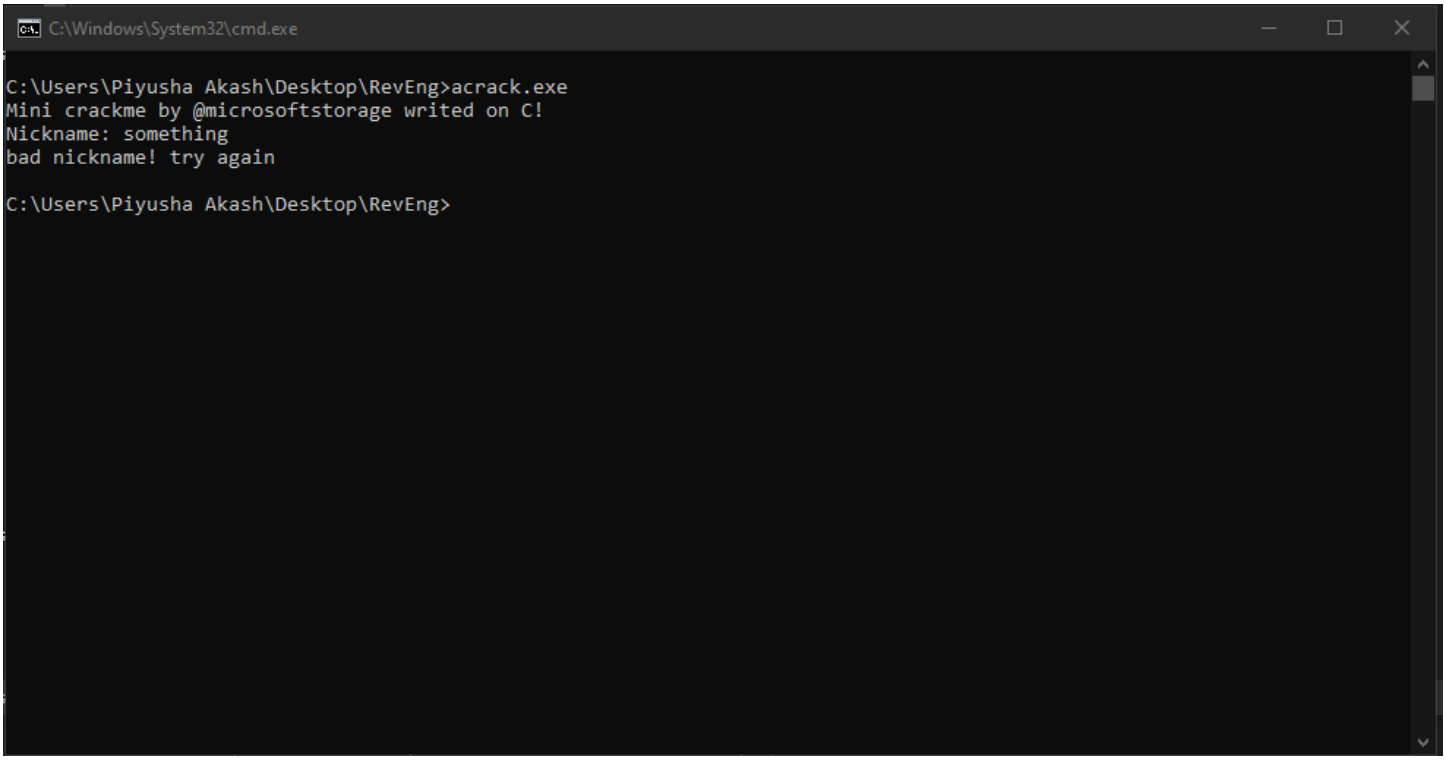
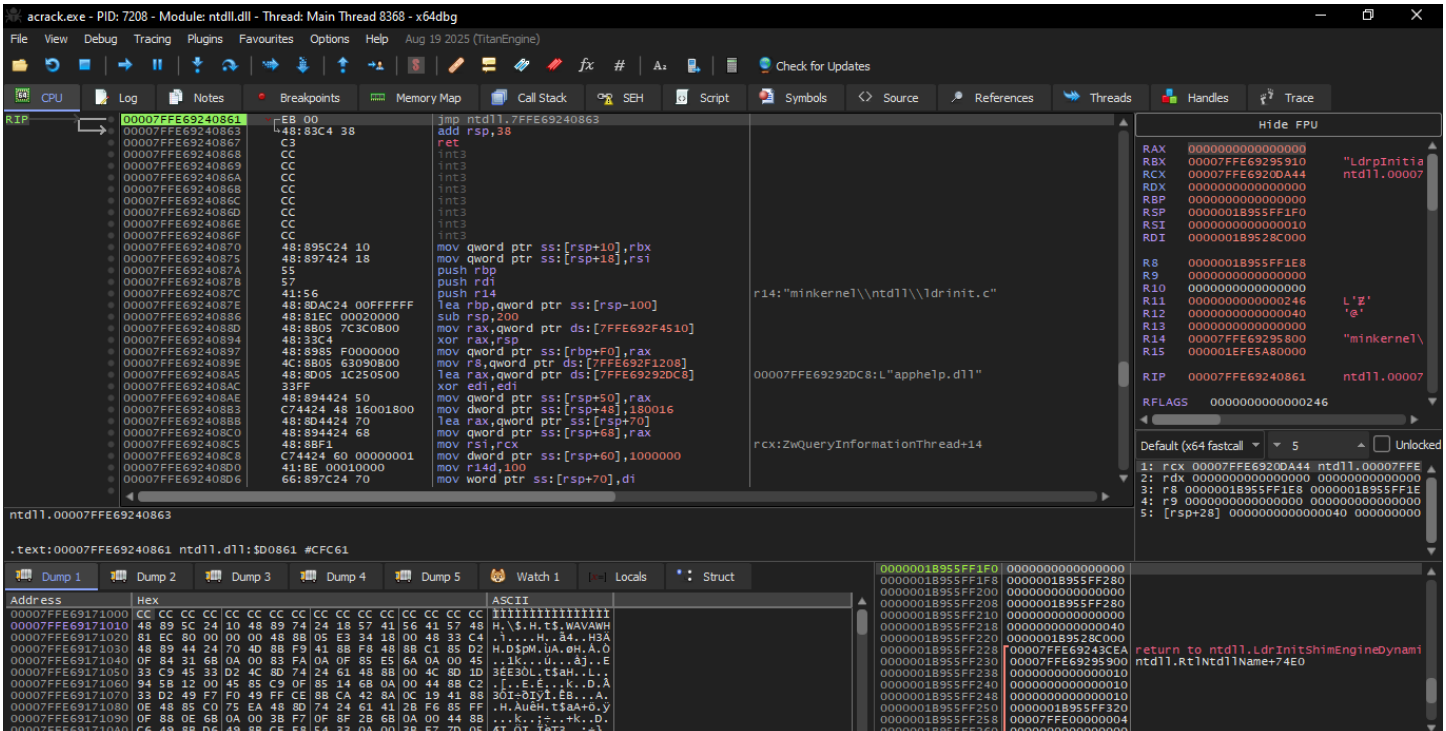
acrack.exe | crackmes.one

EASY LEVEL REVERSE ENGINEERING

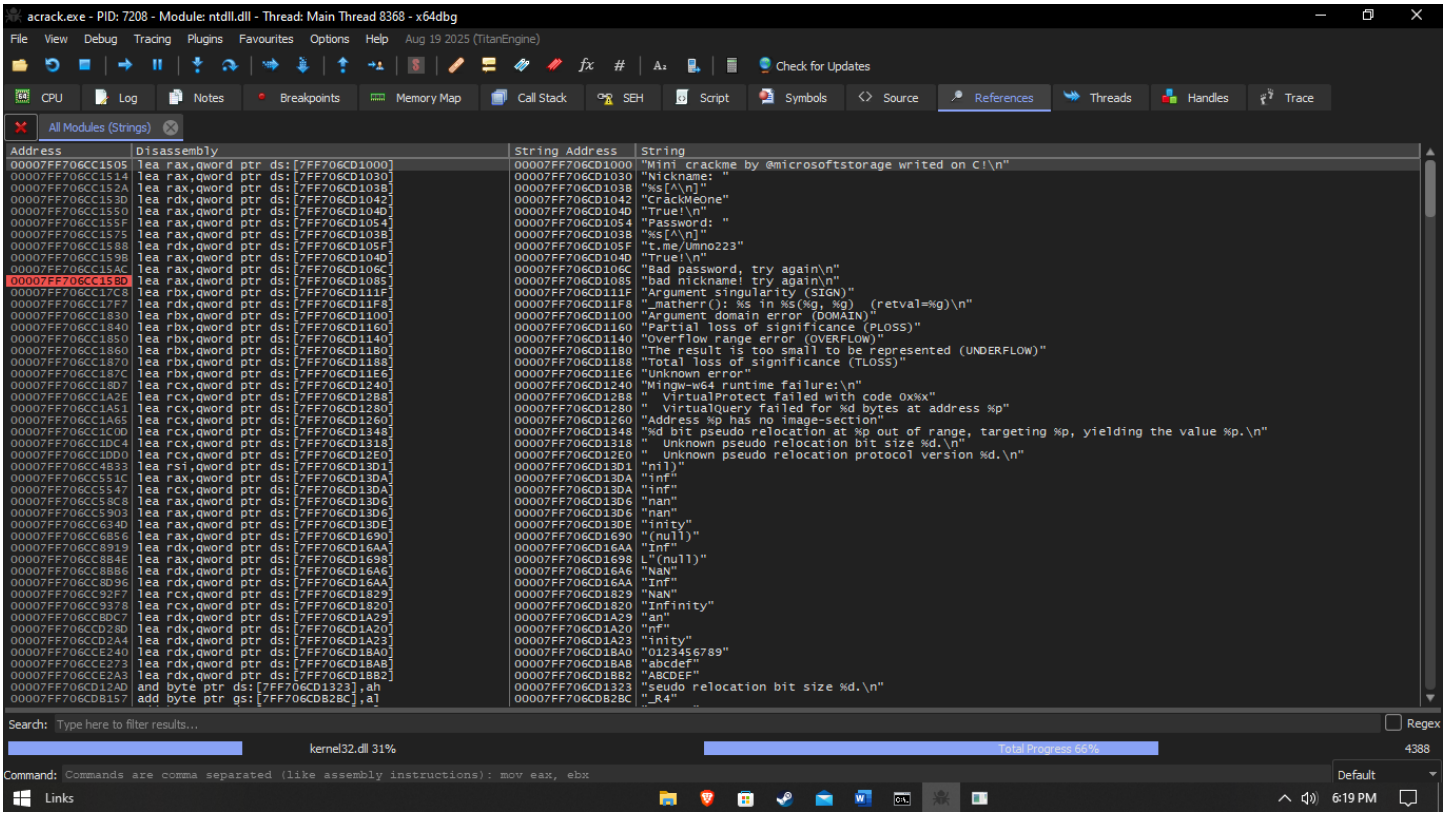
So, this a very easy level RE practice for Reverse engineers Based in C. First of all, we want to see what's the application do. So, I just executed it through cmd.



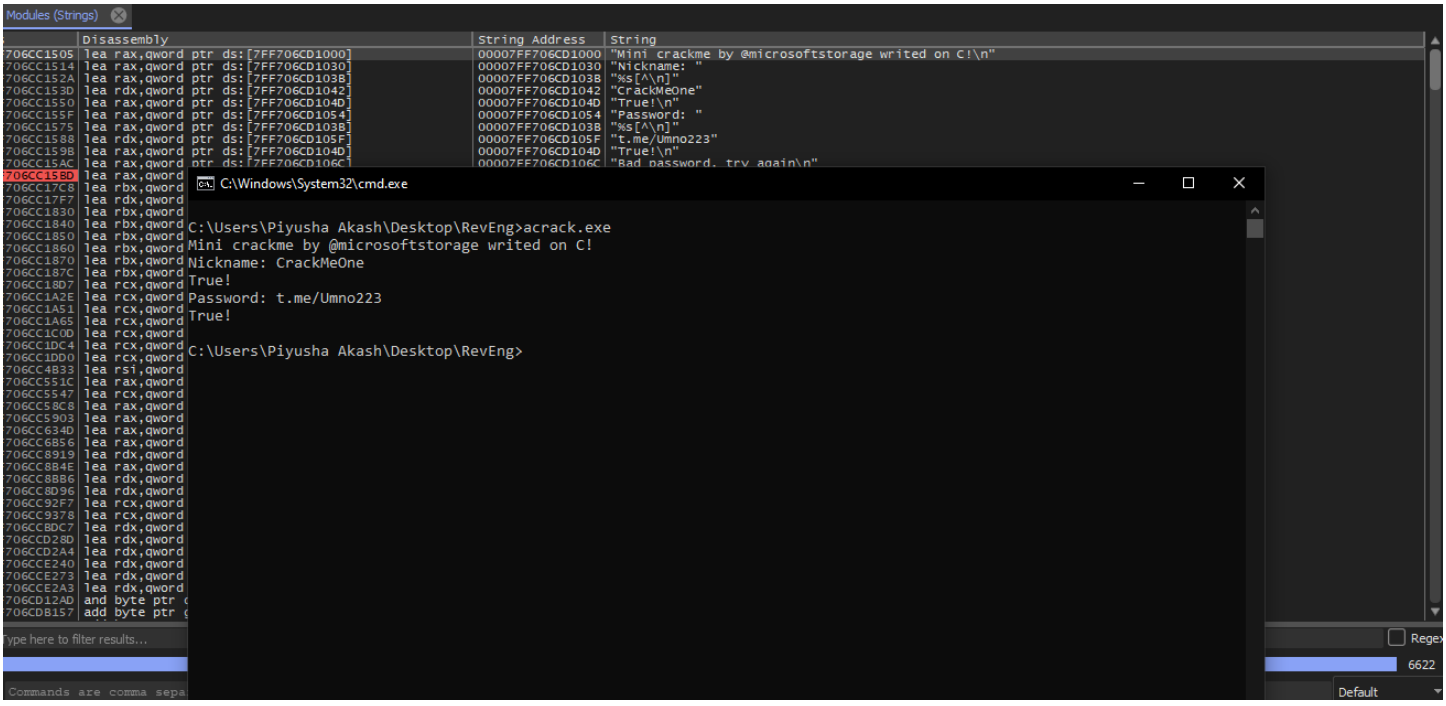
After executing the application, we can see it will ask for Nickname and I just entered "something" as the Nickname and it will give me "bad nickname! Try again." So, we can see it is the wrong Nickname. Let's start Reverse engineer this application. So, in that case I use x64dbg as my tool for reverse this application. I just drag and drop the application to x64dbg.



So, Now I need to looking for strings in this application. So, I will right click the assembly area and go as this. Right click → Search for → All modules → String references. So, I can now see the all strings in this application.

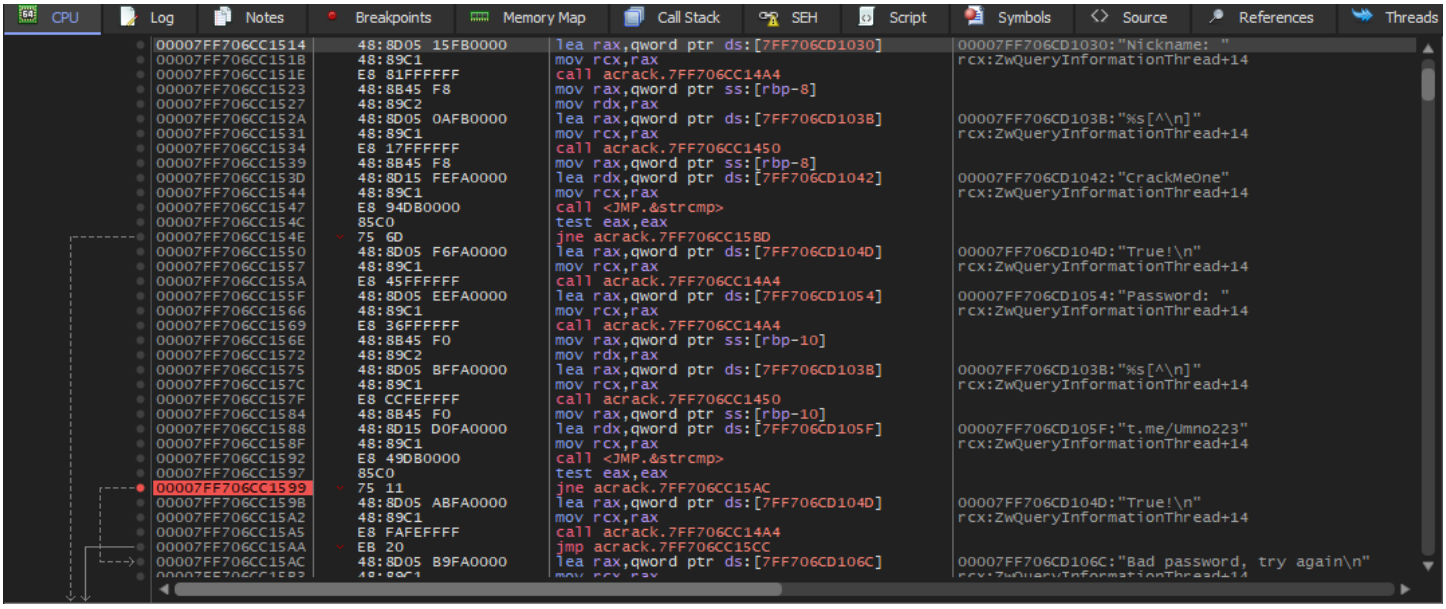


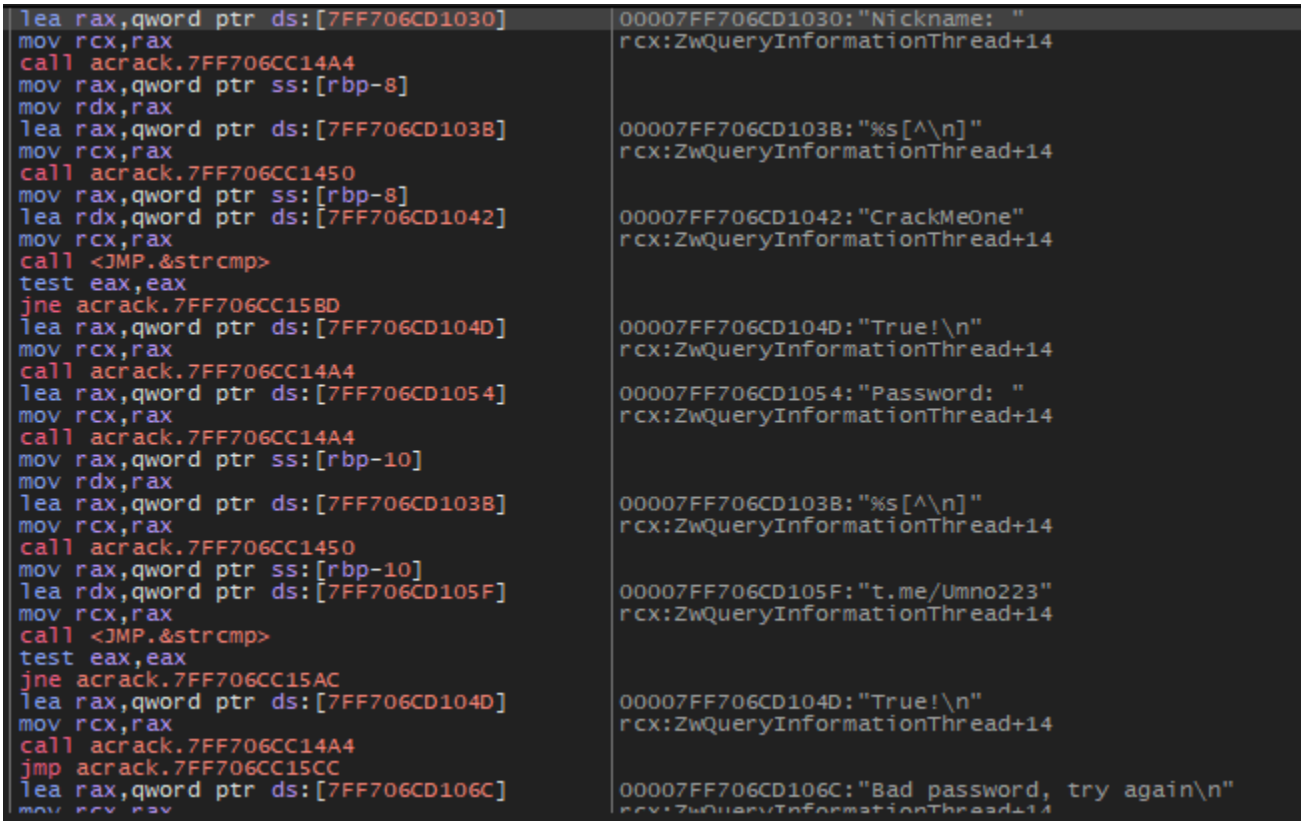
In this section I can see all the string and I will be looking for some hint. And I saw something. I found the password though strings easily. Let's try these credentials.



So, we can see Nickname is CrackMeOne and Password is t.me/Umno223. These credentials are correct and we will get True! Message.

We found the password and nickname but it's not the game. What happen if we can crack the program for return 0 (True! message) for any string. Let's start. So, I just double click the "Nickname" string and then I will go to disassembled section matching to String.

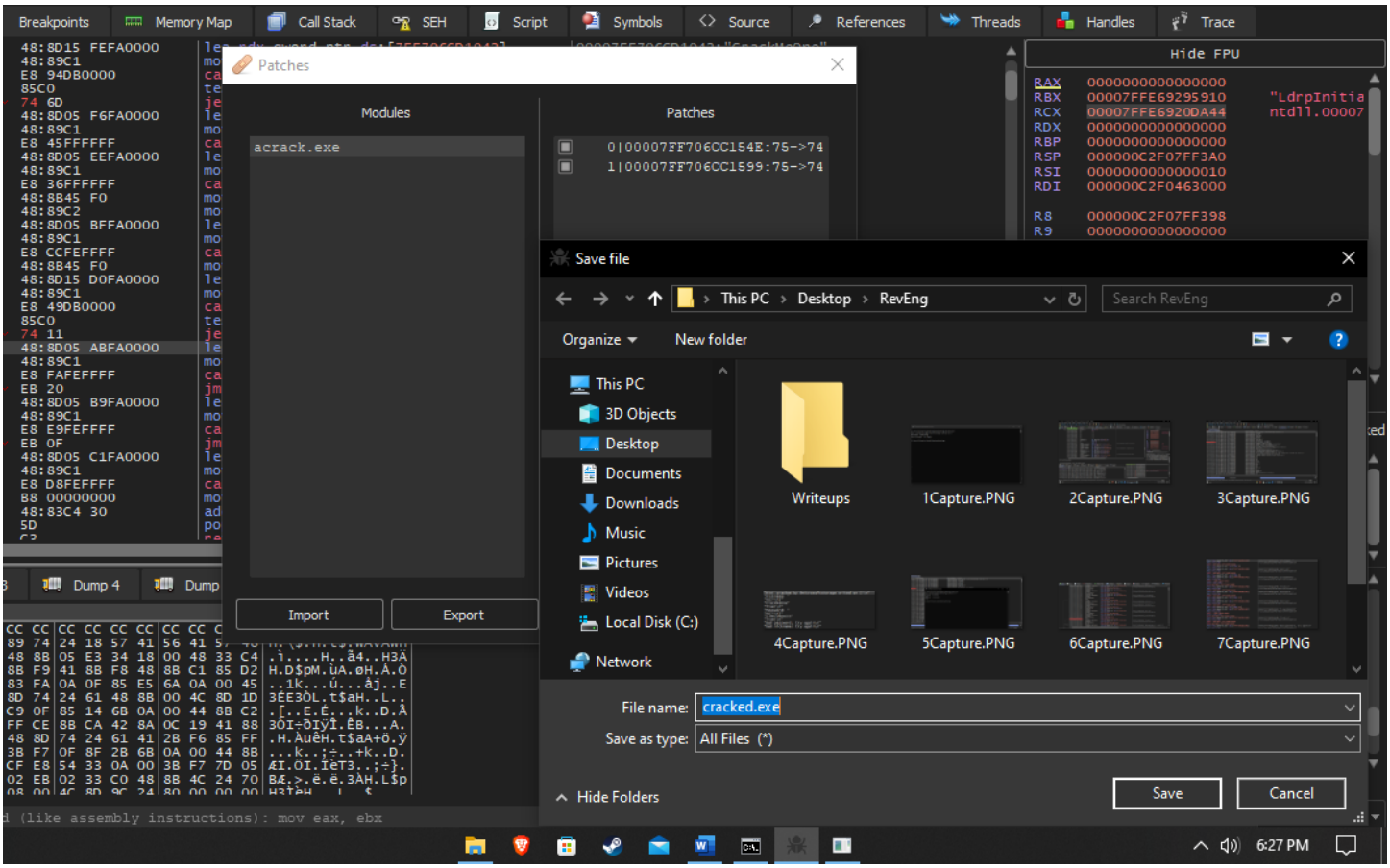




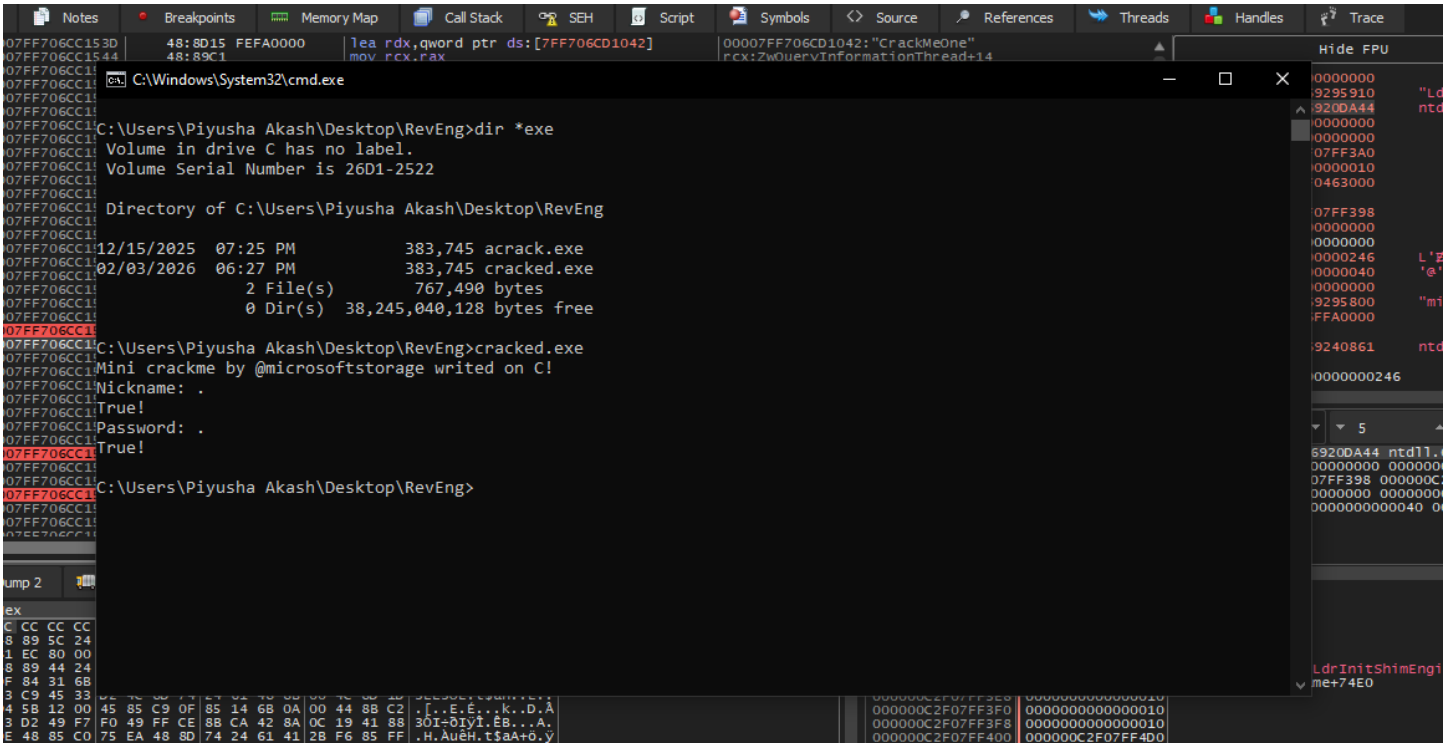
So now I can see Assembly of the program in disassembled section. We need to read this section to understand what's the program do under the hood. So, we can see our password named "CrackMeOne" and in the disassembled section we can see something like "**lea rdx, qword ptr ss::[rbp-8]**". Under this line we can see "**jne acrack.7FF706CC15BD**". So, **JNE** means in assembly is **Jump if Not Equal**. We can see same for our Password.

So that's meaning is If our password doesn't match, Jump to "**acrack.7FF706CC15BD"**. So, let's change the condition. What if we can change the **JNE** to **JE**? **JE meaning is Jump If Equal**. So will go to that line and press space button to assemble that line.

```
lea rdx,qword ptr ds:[7FF706CD1042]      00007FF706CD1042:"CrackMeOne"
mov rcx,rax                              rcx:ZwQueryInformationThread+14
call <JMP.&strcmp>
test eax,eax
je acrack.7FF706CC15BD
lea rax,qword ptr ds:[7FF706CD104D]      00007FF706CD104D:"True!\n"
mov rcx,rax                              rcx:ZwQueryInformationThread+14
call acrack.7FF706CC14A4
lea rax,qword ptr ds:[7FF706CD1054]      00007FF706CD1054:"Password: "
mov rcx,rax                              rcx:ZwQueryInformationThread+14
call acrack.7FF706CC14A4
mov rax,qword ptr ss:[rbp-10]
mov rdx,rax
lea rax,qword ptr ds:[7FF706CD103B]      00007FF706CD103B:"%s[^\n]"
mov rcx,rax                              rcx:ZwQueryInformationThread+14
call acrack.7FF706CC1450
mov rax,qword ptr ss:[rbp-10]
lea rdx,qword ptr ds:[7FF706CD105F]      00007FF706CD105F:"t.me/Umno223"
mov rcx,rax                              rcx:ZwQueryInformationThread+14
call <JMP.&strcmp>
test eax,eax
je acrack.7FF706CC15AC
lea rax,qword ptr ds:[7FF706CD104D]      00007FF706CD104D:"True!\n"
mov rcx,rax                              rcx:ZwQueryInformationThread+14
call acrack.7FF706CC14A4
jmp acrack.7FF706CC15CC
lea rax,qword ptr ds:[7FF706CD106C]      00007FF706CD106C:"Bad password, try again\n"
mov rcx,rax                              rcx:ZwQueryInformationThread+14
call acrack.7FF706CC14A4
jmp acrack.7FF706CC15CC
lea rax,qword ptr ds:[7FF706CD1085]      00007FF706CD1085:"bad nickname! try again\n"
mov rcx,rax                              rcx:ZwQueryInformationThread+14
call acrack.7FF706CC14A4
mov eax,0
add rsp,30
pop rbp
ret
```



So, I just re assembled the **JNE** to **JE** both of lines. And now need to patch this application. So, I will press **CTRL+P** to patch the application. Then patch the application and save new application as **cracked.exe.** Let's execute out new program.



So, as you can see, we will successfully patch the application. Now it will work for any credentials. In that case I put "." As my Nickname and Password. And it's working now. So, what we did? What happened?

We will change the logic in assembly. In assembly this happened. If we entered wrong credentials, Logic jumps to bad credentials message. If it is correct, Logic goes normally. That's why we change **JNE** to **JE**.

Thank you for reading this writeup. Don't forget to follow me. Best regards, **Piyusha Akash. (0x3xp)**

LinkedIn: https://linkedin.com/in/piyushaakash

GitHub: https://github.com/0x3xp

YouTube: https://youtube.com/@infoseclk

TikTok: https://tiktok.com/@infoseclk

Linktree: https://linktr.ee/piyushaakash